New Release of Field Encapsulation Library

Patrick Moran, Chris Henze, Steve Bryson, David Kenwright

The second generation of a software library supporting work related to the design and evaluation of new aircraft has been developed at Ames Research Center. The Field Encapsulation Library (FEL) is designed to support the rapid development of applications involving both computational and experimental fluid dynamics data. The library contains many new features, including a derived-field capability and added support for mesh-independent algorithm development. The new library design also employs a technique known as "lazy evaluation" that makes it easier for the user to write applications that work with large data sets.

The derived-field functionality in the FEL facilitates the creation of new fields in terms of existing ones. For example, many computational fluid dynamics (CFD) solver applications output fundamental solution variables, such as density, momentum, and energy. A typical derived field would be pressure, which can be computed in terms of the fundamental values. There are many derived fields that the user could be interested in viewing. For instance, in the CFD postprocessing application PLOT3D there are over 50 predefined derived fields. The FEL derived-field capability supports all the standard derived fields of PLOT3D; in addition, the library makes it easy for the application programmer to develop custom derived fields.

Another new feature in the FEL is its support for gradient, divergence, and curl differential operators. The built-in support for these commonly used operators makes it easier for users to quickly develop new applications, since more of the necessary field functionality can be used "off the shelf" from the FEL. The library supports computing differential-operator results by either first-order or second-order techniques.

Both the differential-operator fields and derived fields in the FEL employ the lazy evaluation technique. With lazy evaluation, field values are computed when requested, rather than in advance. Lazy evaluation is particularly useful in cases in which an application needs derived values only within small subregions of a field. Many standard visualization techniques, such as streak-line advection or displays on an aircraft surface, fall into this category. The advantages of lazy evaluation are savings in both computation and memory use. Since derived values are only computed on demand, the library never computes values that go unused. Furthermore, no memory is allocated for storing derived values. The benefits of lazy evaluation are especially apparent when working with large time-series data sets, where the drawbacks of precomputing derived values are multiplied by the number of time steps in the data.

Point of Contact: P. Moran
(650) 604-1660
pmoran@nas.nasa.gov